DOI: 10.15439/2014F323
ACSIS, Vol. 2

LELA - A natural language processing system for Romanian tourism

Bernadette Varga*, Alina Dia Trambitas-Miron*, Andrei Roth*,
Anca Marginean†, Radu Razvan Slavescu†, Adrian Groza†
*Semantic Web Department, Recognos Romania
{bernadette.varga, dia.miron, andrei.roth}@recognos.ro
†Intelligent Systems Group
Department of Computer Science, Technical University of Cluj-Napoca
{Anca.Marginean, Radu.Razvan.Slavescu,Adrian.Groza}@cs.utcluj.ro

***Abstract*—This paper presents a commercial semantic-based system for the Romanian tourism. The Lela system exploits both open linked data from Romanian and international sources, and also proprietary databases in the tourism domain. We present the process of creating the linked data set, based on: i) engineering the LELA Romanian tourism ontology, and ii) populating the ontology by linking open data. The system also provides a natural language interface for the Romanian language. The queries are automatically translated into SPARQL based on a controlled vocabulary derived from the Lela ontology.**

***Index Terms*—Semantic information retrieval, Query interfaces, Natural language processing, Linked Data, Tourism ontology**

I. Introduction

LELA is an intelligent blogging-platform designed for providing personalized information about Romanian touristic places. The user can query both subjective and objective information about places of interest. This is possible because Lela uses a custom made semantic annotation tool for blog posts, that identifies points of interest(POIs) and extracts their features and the sentiments expressed about them. The extracted data is used to annotate posts thus allowing their semantic indexing. Lela also provides a Natural Language Question Answering mechanism that allows users to express queries in Romanian language.

II. System architecture

The Lela system relies on the Lela ontology that we engineered for the Romanian touristic domain. The ontology is automatically populated using two methods: i) linking structured data from various sources in the touristic domain, and ii) using natural language processing of available touristic blogs. In the architecture of the system (figure 1) the *Data Collector* module is responsible for the first task, while the *Data Extractor* structures information from blogs in Romanian language. The *Question Answering* module handles queries in natural language against the assertions in the Lela ontology.

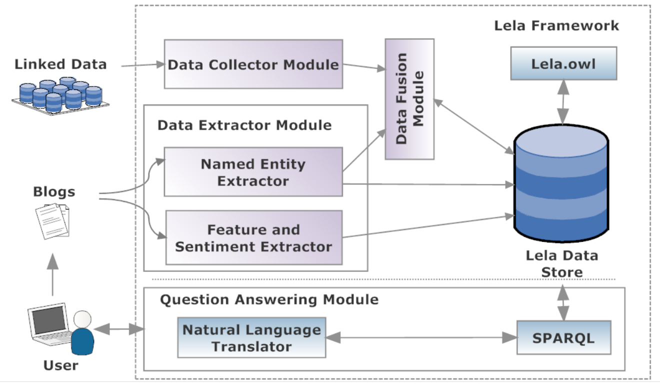


Fig. 1. LELA system architecture.

The *Data Collector* module identifies and imports relevant information related to Romanian points of interests by linking touristic information from open data provided by the Romanian agencies, complemented with relevant knowledge from Wikipedia, DBpedia, or Freebase. Data is collected using available SPARQL endpoints or source specific APIs. As information related to the same POI is usually available in more than one data source, we have developed a customized *Data Fusion* module, aiming to identify individuals described in different data sets. Based on information such as geospatial location and textual description, the equivalent individuals are linked via the *same-as* relationship and the various information asserted is fused.

Lela framework provides also a blogging platform where Romanian bloggers can write their stories about the places they have visited. In order to respond to the specific needs of bloggers that already built a readers community and an online reputation via their own blogs built on platforms such as WordPress, Drupal, Blogger, Tumblr, *etc.*, we will provide custom plug-ins for each one of the systems mentioned above, that will allow the semantic indexing of blog posts. Even though the content published on those blogs will not be copied on our platform, it will be available for Lela users to query and explore. This way we will acquire subjective information (stories, opinions) related to POIs.

The *Data Extractor* module analyses the available blogs in order to: i) perform Name Entity Recognition for the main concepts in the Lela ontology: points of interest, accommodation, restaurants or touristic activities; ii) assert different relationships between individuals in the ontology, as they appear in

978-83-60810-58-3/$25.00 © 2014, IEEE

the blogs, and iii) identify sentiments expressed in relation to each feature of a specific concept in the ontology. For instance, for the `Accommodation` concept, we are interested in opinions regarding features like `location`, `view`, `comfort`, `furniture`, `service` or `value for money`. The facts corresponding to these features (e.g., `Hotel X` serves `good food`) are stored into the ABox using the `Match` concept, which references the blog post that was analyzed, and the position inside the text for the POIs and opinions that were identified("matched") by the system.

For opinion detection, we used a machine learning technique based on Conditional Random Fields (CRFs) [1]. We employ this technique in order to find an appropriate labeling for blog sentences, regarded as sequences of words. The labels we try to detect describe the position in a sentence of a word which refers a specific instance of a concept (e.g. "Grand Hotel Italia"), a specific feature we are interested in (e.g. "cazare (accommodation)") and the associated opinion (e.g. "bun (good)"). For opinion polarity, we used the WordNet-Affect for Romanian [2].

The module which labels the text uses a model generated in the training phase, starting from a set of 200 manually labeled phrases. This set is further expanded by replacing some words of interest (especially the opinion adjectives like "good") with their synonyms, thus obtaining more training examples. A set of attributes has been selected for describing each word in the training set, and among them are the word's Part-Of-Speech, whether the word belongs to an entity of interest and the type of entity. For example, let us consider the sentence "Am fost la Transilvania International Film Festival si mi-a placut" (I was at the Transilvania International Film Festival and liked it). The attributes associated with the word "Transilvania" will have the following values: "NNP" for the Part-Of-Speech, "B" for the attribute which specifies the name of the entity starts here and "EVB" for the attribute specifying the word starts the name of an event.

The model generated based on the training example could be improved by expanding the set of examples and/or attributes. A separate module allows adding new attributes ("features" in CRF terminology) and computing their corresponding values before generating the new model. Once this is done, the new model is used to detect the entities the text is talking about, their specific features and the opinion on them. The opinion information gets stored in the A-Box as explained above and can refer either a feature of a concept instance or a pair Activity-Location (e.g., "skiing" in "Predeal").

The opinions concerning each feature of a specific instance are aggregated into a quality score for that particular feature. The function which performs this takes into account both the detected opinion polarities (on a scale from -2/very bad to +2/very good) and the weights specified by the user for each feature s/he might be interested in, according to their importance from his/her point of view. When the discovered Named Entities are not recognized as Romanian POIs available within the Lela Data Store, they are added to the data store as new instances.

```
11. (define-role fromBlogPost :domain Match :range BlogPost)
12. (define-role hasSubject :domain Match :range LelaAxis)
13. (define-concrete-domain-attribute hasScore :domain Match
:type real)
14. (define-concrete-domain-attribute hasText :domain Match
:type string)
15. (define-role speaksAbout :domain BlogPost :range
LelaAxis)
16. (instance m1 Match)
17. (instance b100 BlogPost)
18. (instance mateicorvin POI)
19. (attribute-filler m1 "casa matei corvin atrage multi
turisti" hasText)
20. (related m1 b100 fromBlogPost)
21. (related m1 mateicorvin hasSubject)
22. (attribute-filler m1 0.8)
```

Fig. 3. Relating information about a blog with the *n-ary design* pattern.

III. Linked data creation process

The process of creating the Lela linked data set consists of three main steps: i) engineering a Romanian tourism ontology, ii) developing of data collection and data fusing modules, iii) publishing the resulting data sets.

A. *Definition of a Romanian tourism ontology*

To develop the Lela ontology, we follow the methodology in [3] and we also enact various ontology design patterns [4].

The later is described in KRSS syntax[1]. The four axes of the Lela-core ontology are `Accommodation`, `Activity`, `EatingAndDrinking` and `PointsOfInterest`, denoted by `POI` (line 1 in figure 2). Apart from those, Lela ontology also offers special classes for describing events, price, infrastructure, contact details, facilities of each point of interest, etc. The main properties defined in our ontology have restricted domains and ranges (figure 2 lines 3-6) which are used to facilitate reasoning among the top level concepts. The partition design pattern [7] was used to partition the top level of the ontology.

Beside the top level concepts, we also introduces the concept `Match` for representing the relations between the touristic places and the blog posts that POIs appeared in. This concept was modelled by enacting the *n-ary ontology design pattern* [7]. The goal was to combine several information about a tourism blog (see fig. 3) regarding: subject of the blog according to the concepts in Lela (line 12), computed score about an instance in the ontology (axiom 13), or provenance information like author, starting and ending text index (text position) which relates to an identified instance in our ontology. As an example, the individual `m1` of type `Match` is related to the blog `b100` via the role `fromBlogPost`.

The point of interest `mateicorvin` is related to the same match `m1` by the relation `hasSubject`. The positive score

[1] For a detailed explanation about families of description logics, the reader is referred to [5], while for the complete KRSS syntax to [6].

```
(define-concept LelaAxis (or Accommodation Activity EatingAndDrinking POI Location))
(disjoint Accommodation Activity EatingAndDrinking POI Location)
(define--role hasAccommodation :domain (or Activity EatingAndDrinking POI):range Accommodation)
(define-role hasActivity :domain (or Accommodation EatingAndDrinking POI) :range Activity)
(define-role hasEatDrink :domain (or Accommodation Activity POI) :range EatingAndDrinking)
(define-role hasPOI :domain (or Accommodation Activity EatingAndDrinking) :range POI)
(define-role hasLoc :domain LelaAxis :range Location :transitive t :inverse LocatedIn)
```

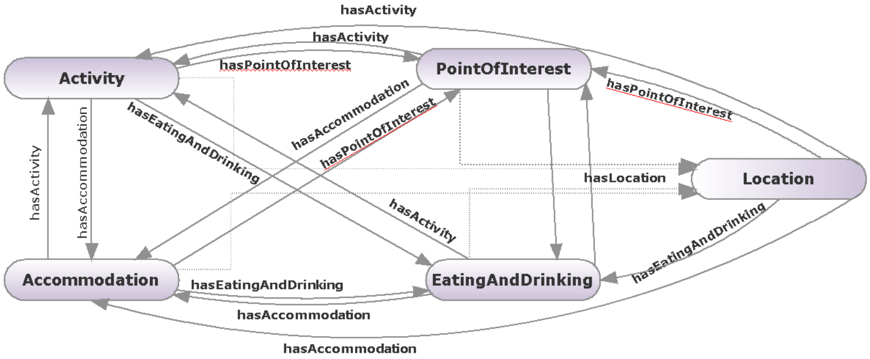


Fig. 2. Top level of the Lela-core ontology module.

of 0.8 in line 22 is computed with a basic opinion mining algorithm from the blog post.

B. Data collection and fusion

We focused on discovering and selecting the relevant open data sets for the Romanian tourism domain. Touristic points of interest are collected from tow sources: i) the available $POI\ data$ sources (Wikipedia, Freebase, DBPedia, Geonames, Wikisherpa, Wikitravel) and ii) Named Entity recognition from touristic blogs. Data fusion is performed in AllegroGraph and saved as a triple store[2], while RacerPro server is used for reasoning on the Lela ontology.

The following sources were exploited: *Wikipedia*, *DBpedia*, *Geonames*, *Freebase*, *Wikisherpa*, *WikiTravel* (table I). A proprietary dataset from Cluj4All (www.cluj4all.com - Recognos' own database about Cluj-Napoca with over 7000 described objectives, from which 1000 relevant touristic points) was also included. The data sets provided by various Romanian governmental agencies were used containing information for Romanian museums, churches, and historical points.

Wikipedia has categorized pages, and some of it's textual content tagged. There are very few consistent patterns followed by the content generators or authors (an exception would be the infobox content in the right side). However we observed that similar tags were used for describing the Romanian touristic objectives, and similar naming conventions for pages. For example, we retrieved values from page that respected the pattern "List_of_places_from_Cityname" from (http://ro.wikipedia.org/wiki/Lista_locurilor_in_Cluj-Napoca).

TABLE I
LINKING AVAILABLE DATASETS.

Data set	Available at	Description
Romania Museum Guides	http://data.gov.ro/dataset	Descriptive data and geolocations of 967 museums in Romania
Wikipedia	http://wikipedia.ro	Various categories about Romanian touristic places
Freebase	http://www.freebase.com/	Community-curated database of well-known people, places and things - some about Romania
Geonames	http://www.geonames.org/	Covers all countries and contains over eight million place names - some about Romania
Wikisherpa	http://www.wikisherpa.com/	Data from wikiTravel in a more structured way
DBpedia	http://dbpedia.org/	Structured data from wiki to other external resources
Cluj4All	cluj4all.com	Around 7000 objectives about Cluj-Napoca

TABLE II
LINKING LELA ONTOLOGY WITH DBPEDIA.

Lela concepts	DBpedia concepts
POI	Museums, Castles, Towers, Churches, Cathedrals, Monuments, OutdoorSculptures, Bridges, Parks, Zoos
Activity	Cinemas, Theater, Activity, Shopping
Accommodation	Hotel
EatingAndDrinking	Restaurant

DBpedia organizes its data into triples, and data is linked to external data sets [8].

We queried the DBpedia database for 5 main cities (Bucharest, Cluj-Napoca, Timisoara, Brasov and Sibiu) following a predefined mapping of the 4 main Lela classes to the DBpedia specific classes (table II). A simple example for such

[2] Available at http://www.recognos.ro/lela/LelaLinkedDataSet.nq

```
SELECT distinct
?subject ?latd ?longd ?about ?image ?category
?sameAs ?abstract ?wikipedia ?label
WHERE {
   ?subject <http://purl.org/dc/terms/subject>
            <http://dbpedia.org/resource:Category:
             Museums_in_#placeName\#>.}
   OPTIONAL {?subject dbpedia-owl:thumbnail ?image.}
   OPTIONAL {?subject rdfs:label ?label.}
   OPTIONAL {{?subject foaf:homepage ?about.}}
   OPTIONAL {{ ?subject geo:lat ?latd. ?subject
               geo:long ?longd.} union
            {?subject dbpprop:latitude ?latd.?subject
               dbpprop:longitude ?longd.}}
   OPTIONAL {?subject owl:sameAs ?sameAs.
            FILTER contains(str(?sameAs), "freebase").}
   OPTIONAL {?subject foaf:isPrimaryTopicOf ?wikipedia.
            FILTER contains(str(?wikipedia), "wikipedia").}
   OPTIONAL {?subject dbpedia-owl:abstract ?abstract.
            FILTER (LANG(?abstract)='ro' ||
                    LANG(?abstract)='en')}
   BIND('Museum' AS ?category).
            FILTER (!contains(str(?subject), "List_of")).
}}
```

Fig. 4. Querying DBpedia for the `Museum` category.

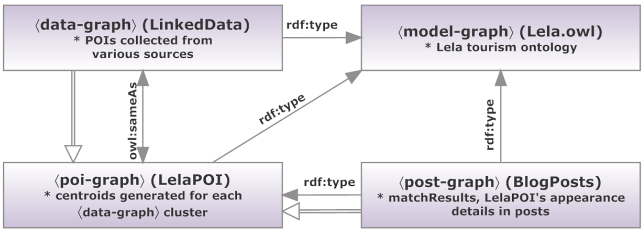

⟨data-graph⟩ (LinkedData)
* POIs collected from various sources
rdf:type
⟨model-graph⟩ (Lela.owl)
* Lela tourism ontology
owl:sameAs
rdf:type
rdf:type
⟨poi-graph⟩ (LelaPOI)
* centroids generated for each ⟨data-graph⟩ cluster
rdf:type
⟨post-graph⟩ (BlogPosts)
* matchResults, LelaPOI's appearance details in posts

Fig. 5. LELA graph relations.

a query is shown in figure III-B, where where we extracted geospatial data, image, category, abstract and Wikipedia link for all instances of the Wikipedia' `Museum` category. *Geonames* has it's own category and ranking system for objectives. We retrieved the Hotels, Restaurants and Point of Interest for the above mentioned cities. *Freebase* and *Wikisherpa* have similar data collection processes. We accessed these resources through their APIs, and stored the obtained values in the Lela data store. Finally, we imported the xml and xls data from the government provided sources, and the Cluj4All private database.

The above cited data sources have many POIs that are relevant for the tourism domain or for Romania in general. However we only considered those POIs that satisfied both constraints at the same time, for the 5 most significant Romanian cities. As a result, the import process collected around 5.000 POIs, but the data was still noisy, as it contained many overlaps in the form of several instances of POIs collected from different sources semantically describing the same object. In order to overcome this issue, a fusion algorithm was applied. In order to make this process easier, we stored the data in several independent graphs, as the triple store solution we used - AllegroGraph - offers the possibility to partition data for an easier management. The reasoning mechanism can also be applied on the graph-level, not taking into account the several other graphs that might exist in a triple store. This proved to be helpful, when dealing with a bigger amounts of data.

In figure 5 the graph partitioning is shown. We created four separate graphs. The ⟨*model-graph*⟩ contains the TBox details of the Lela ontology. It includes the tourism taxonomy concepts and relations between them, as well as the object and data properties. The ⟨*data-graph*⟩ contains the external data, imported as triples. This graph uses some properties defined by the ⟨*model-graph*⟩ (concepts, label properties for names etc). In the fusion process, in the ⟨*data-graph*⟩ a representative element is created for each group of objectives semantically describing the same element. These groups are called by us POI clusters, and the representative element is the centroid POI. These centroids are saved separately in the ⟨*poi-graph*⟩, and we call them LelaPOIs. They have their derived properties from the cluster data, they also have a `owl:sameAs` property referring their original sources (instances from ⟨*data-graph*⟩). This process supports more efficient query plans, that exploit data from remote sources only when additional information related to a given POI is explicitly requested. Because of the `owl:sameAs` property, AllegroGraph allows us to get any details from data graph. The fourth graph is the ⟨*post-graph*⟩, which contains the blog post related information, like the a blog post's content or the matching details, after the post analysis.

The corresponding algorithm is summarized in Algorithm 1. In a first step, the algorithm finds all the instances that have the same wikipediaUrl, and link them with `owl:sameAs` property. The wikipedia urls are not ambiguous, so the operation will be correct. In a second step, it finds all the instances that have the same freebaseUrl, and if there is `owl:sameAs` between them, then add it.

Thirdly, it checks equality of label values(compare lela:hasName properties). If perfect match found and the objects are located in the same place, and no equality still reported, then link them with `owl:sameAs`.

Fourthly, the algorithm finds all the instances that have the same Web page. If there is no `owl:sameAs` link between them, adds it. Two additional steps have been also applied:

1) *Fusion cities* that have been read and imported from an xls document with cities and instances that have been imported from other resources like Geonames, Freebase.
2) *Fusion counties* that have been read and imported from an xls document with counties/instances that have been imported from other resources (Geonames, Freebase or others) - based on the previously mentioned information and same `lela:hasName` property.

Finally, based on the previously generated groups (a group is considered as a series of elements related by the `owl:sameAs` property) a special instance for each cluster is created in ⟨*poi-graph*⟩ (recall figure 5). The centroid of the group is asserted as an instance of the corresponding most specific concept from the Lela ontology.

```
Data: $KB$, the LELA Knowledge Base;
      $xlsCities, otherCities$, lists of cities;
      $xlsCounties, otherCounties$, lists of counties;
      $poi$, the LELA POI graph;
Result: an augmented LELA Knowledge Base
foreach $i \in instances(KB)$ do
    foreach $j \in instances(KB)$ do
        if $i \neq j$ then
            if $wikipUrl(i) = wikipUrl(j) \vee freebaseUrl(i) = freebaseUrl(j)$ then
                $assert($owl:sameAs$(i, j), KB)$
            end
        end
    end
end
foreach $i \in instances(KB)$ do
    foreach $j \in instances(KB)$ do
        if $i \neq j$ then
            if $lela{:}hasName(i) = lela{:}hasName(j) \wedge loc(i) = loc(j)$ then
                $assert($owl:sameAs$(i, j), KB)$
            end
        end
    end
end
foreach $xc \in xlsCities$ do
    foreach $oc \in otherCities$ do
        if $xc \neq oc$ then
            if $wikipUrl(xc) = wikipUrl(oc) \vee freebaseUrl(xc) = freebaseUrl(oc)$ then
                $assert($owl:sameAs$(xc, oc), KB)$
            end
        end
    end
end
foreach $xc \in xlsCounties$ do
    foreach $oc \in otherCounties$ do
        if $xc \neq oc$ then
            if $(lela{:}hasName(xc) = lela{:}hasName(oc)) \wedge (wikipUrl(xc) = wikipUrl(oc) \vee freebaseUrl(xc) = freebaseUrl(oc))$ then
                $assert($owl:sameAs$(xc, oc), KB)$
            end
        end
    end
end
$clusters \leftarrow Partition(instances,$owl:sameAs$)$
foreach $c \in clusters$ do
    $i \leftarrow selectSpecialInstance(c)$
    $addToGraph(i, poi)$
end
```

Algorithm 1: LELA fusion algorithm

```
31. cat: EatingandDrinking, Location,
          Accommodation, POI, Activity,....;
32. fun ActivityhasLocation :
    EatingandDrinking -> Location -> PropertyCl;
33. Pizza : Object;
34. VSki: ActivityVerbPhrase;
35. VDrink, VEat : EatingandDrinkingVerbPhrase ;
36. V2Eat : Object ->
      EatingandDrinkingVerbPhrase;
37. QWhereModVerbPhrase :
      Modality -> VerbPhrase  -> Question;
```

Fig. 6. Abstract grammar derived from the Lela ontology.

C. Saving and publishing the resulting data sets

The data collection process resulted in the import of approximately 5.000 instances, some of them semantically describing the same point of interest, without any flag pointing out their equality. To eliminate this issue, the instances were grouped into clusters, based on characteristics such as their names, wikipedia pages, spatial coordinates, etc. For each cluster, the centroid was selected to became an instance of the `LelaPOI` concept. The centroid and the other individuals in the cluster are linked via a specific `similarity` relationship asserted in the LELA ontology. The unified data set is stored in a local AlegroGraph triplestore [9]. Currently the triplestore contains around 3.200 unique tourism objectives collected for five cities. The points of interest are described by 40.697 of RDF triples.

IV. Querying the linked data set in controlled language

To explore the linked dataset we provide a natural language query interface. The queries can be expressed in a controlled vocabulary for the Romanian language. The queries in natural language are automatically translated into SPARQL. The translation is based on three grammars that we developed in the Grammatical Framework [10], [11]:

1) one abstract grammar, derived from the Lela ontology;
2) one concrete grammar for the Romanian language
3) one concrete grammar for the SPARQL.

First, the abstract grammar in figure 6 is based on the main concepts and roles of the Lela ontology. The concepts in Lela are represented as categories in the grammatical framework, while roles as functions (lines 31-32). Individuals in the ontologies are modelled as instances of generic type *Object* (line 33). Activities are encapsulated as VerbPhrases (i.e., the verb VSki for the ski activity in line 35). Various eating and drinking activities are modelled with a specific verb phase (i.e., *EatingandDrinkingVerbPhrase* in line 36). The function introduces in line 36 is used to represent eating and drinking activities with parameters (i.e., eating pizza). The query template in line 37 is used to match against queries which include modal verbs (i.e., where can I eat pizza?).

Second, the concrete grammar for the Romanian language (figure 8 defines the controlled natural language used to query the system. The relevant verbs in the tourism domain are

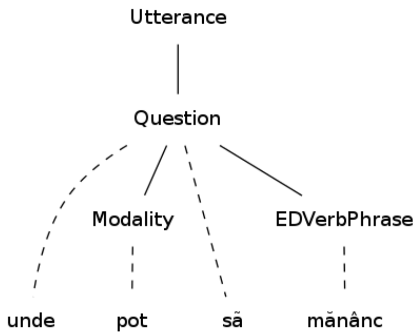


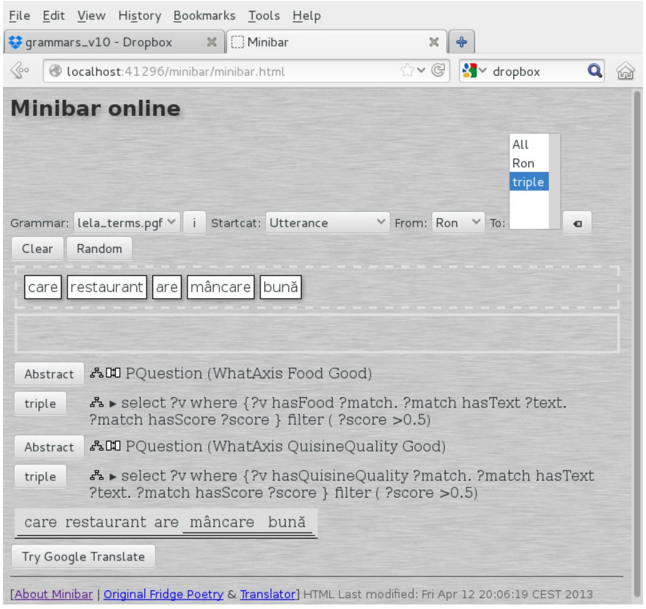
Fig. 7. Parse tree for ”Where can I eat?”. The pronoun is missing because the flexional form of the predicate `EDVerbPhrase` is enough to deduce the agent of the query.

```
> p "unde putem sa jucam  badminton"
     |l -lang=triple
SELECT ?where, ?activity where {
 ?activity rdf:type ex:BadmintonActivity.
 ?where ex:hasActivity ?activity .
 ?match rdf:type Match . ?match hasSubject
 ?where. ?match hasScore ?score }
ORDER by Desc(?score)

>p "unde pot sa schiez" |l -lang=triple
SELECT ?where, ?activity
WHERE { ?activity rdf:type ex:SkiActivity.
      ?where ex:hasActivity ?activity .
      ?match rdf:type Match .
      ?match hasSubject ?where.
      ?match hasScore ?score }
ORDER by Desc(?score)

>p "unde este restaurantul Agape"
    |l -lang=triple
SELECT ?location where
    { ex:id342451 ex:hasLocation ?location}
```

Fig. 10. Queries in Romanian language automatically translated in SPARQL.


File Edit View History Bookmarks Tools Help
grammars_v10 - Dropbox
Minibar
localhost:41296/minibar/minibar.html
dropbox
Minibar online
All
Ron
triple
Clear
Random
Try Google Translate
[About Minibar | Original Fridge Poetry & Translator] HTML Last modified: Fri Apr 12 20:06:19 CEST 2013

Grammar: lela_terms.pgf i Startcat: Utterance From: Ron To: triple

care restaurant are mâncare bună

Abstract PQuestion (WhatAxis Food Good)

triple select ?v where {?v hasFood ?match. ?match hasText ?text. ?match hasScore ?score } filter (?score >0.5)

Abstract PQuestion (WhatAxis QuisineQuality Good)

triple select ?v where {?v hasQuisineQuality ?match. ?match hasText ?text. ?match hasScore ?score } filter (?score >0.5)

care restaurant are mâncare bună

Fig. 11. Qualitative queries filter the results based on the available opinions on the topic of the query.

specified (like drink in line 1, eat in line 2, sky in line 3) v_beschX are functions for smart paradigms of a language that provide different inflexion of verbs for different persons, numbers and tenses for the Romanian language. Romanian specific parsing rules are used here to define equivalence between related queries. For instance, specific to the Romanian language is to commit the pronoun in questions like ”Unde pot să mănânc?” (Where can I/We/You eat?), instead of ”Unde pot eu să mănânc?” (Where can I eat?) we define new forms for pronouns `ifemAbsent_Pron` (lines 44-48). The corresponding parse tree is depicted in figure 7, where the personal pronoun does not appear for the Romanian version of the query, as it can be deduce from the verb’s flexional form.

Thirdl, the concrete grammar for SPARQL was develop to automatically translate the controlled natural language into a formal query. GF uses the grammars both for parsing and linearizing, therefore a translation from a Romanian phrase to a SPARQL query is done by (1) parsing in the grammar of Romanian language followed by (2) linearising the obtained parsed tree in SPARQL concrete grammar. For each verbal phrase, we give the corresponding SPARQL statement together with the name of the variable that is to be included in the SELECT clause of the query. The grammar in figure 9 is used to translate questions related to eating, drinking or other touristic activities (such as `SkiActivity`), queries which include modal verbs. The SPARQL grammar is exemplified in figure 10. The three queries in Romanian language illustrated in figure 10 are: ”Where can we play badminton?”, ”Where can I ski?”, ”Where is the Agape restaurant?”. The resulted SPARQL code can be used to directly query the Lela ontology

The system allows *qualitative queries* i.e., ”What is the atmosphere at Pizzeria Napoli?”, ”Which is the best restaurant with Romanian cousin?”. Answering to these queries exploits the knowledge provided by the opinion analyser module. The qualitative query ”Which restaurants have good food?” in figure 11 is matched against two concepts in the Lela ontology: i) intersection between `Food` and `Good` and ii) the concept defined by those instances whose role `QuisineQuality` points towards the concept `Good`. The corresponding SPARQL queries filter the results to those instances classified by the opinion analyser module as positive (that is $?score > 0.5$).

The Romanian grammar is used to generate all the flexional forms of the vocabulary in order to guide the user to generate grammatically correct queries (12). In our case, the vocabulary is restricted to touristic terms from the Lela ontology. After typing a word, the system displays all the possibilities to complete the question in the defined controlled natural language.

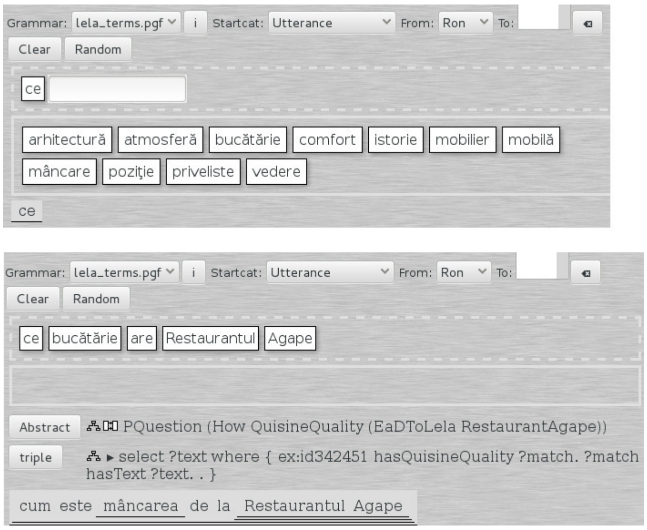
To sum up, the translator of from Romanian language to SPARQL is able to handle the following types of queries in

```
VDrink = mkVP (v_besch73 "bea");
VEat   = mkVP (v_besch52 "manca");
VSki   = mkVP (v_besch10 "schia");
QWhere&ModVerbPhrase m vp =
  mkQS (mkQCl where_IAdv (mkCl (mkNP ifemAbsent_Pron) (mkVP m  vp)))
 |mkQS(mkQCl where_IAdv (mkCl (mkNPweAbsent_Pron) (mkVP m vp)))
ifemAb&sent_Pron =
P.mkPronoun [] "mine" "mie" [] [] "meu" "mea" "mei" "mele" Fem Sg P1 ;
```

Fig. 8. Concrete grammar for the Romanian language.

```
VDrink = {v="?eatdrink"; body="?eatdrink rdf:type ex:EatingandDrinking."};
VEat =   {v="?eatdrink"; body="?eatdrink rdf:type ex:EatingandDrinking."};
VSki =   {v="?activity"; body="?activity rdf:type ex:SkiActivity."};
QWhere&EatingandDrinkingDModVerbPhrase  x y ="select "++ y.v ++ "where {"
       ++ y.body ++
       ++ ``?match rdf:type Match . ?match hasSubject "++ y.v ++"."
       ++ "?match  hasScore  ?score }"
       ++"ORDER by Desc(?score)";
```

Fig. 9. Part of the grammar developed to translate a query into SPARQL.



PQuestion (How QuisineQuality (EaDToLela RestaurantAgape))

select ?text where { ex:id342451 hasQuisineQuality ?match. ?match hasText ?text. . }

cum este mâncarea de la Restaurantul Agape

Fig. 12. Guiding the process of constructing queries: (top) after the word ”ce” (what/how) is types only the grammatically correct flexional forms remain (bottom) the SPARQL version for the query ”How is the food at Agape restaurant?”.

which for each type several linguistic patterns are modelled: 1) Retrieving location of various elements from the Lela ontology (Accommodation, Eating and Drinking, Activities, POIs, etc.); 2) Identifying and describing simple activities (swim, walk) and compound activities (play badminton); 3) Handling queries containing reflexive or verbs with direct object; 4) Handling questions in which the subject is not explicitly expressed; 5) Enhancing verbs with modalities (can, should, may, etc.); 6) Qualitative queries.

V. Discussion and related work

Natural Language to SPARQL. To our knowledge, this is the first system which translates queries from the Romanian language into SPARQL syntax. The system relies on a domain-dependent controlled vocabulary in the tourism domain. For the English language, various systems do exist [12].

The *QuestIO* system [13] is open-domain, with the vocabulary automatically derived from the data existing in the knowledge. The system was designed to handle language ambiguities and incomplete or syntactically ill-formed queries by enacting fuzzy string matching and ontology similarity metrics. We focused on several specific difficulties for the Romanian language like: i) the inconsistent use of diacritics and special symbols, or ii) the flexibility of the sentence structure, which allows questions with or without pronouns.

The ONLI+ system [14] is a portable ontology-driven question answering system for English language. Similar to our work, the RacerPro system was used to reason on the ontology and to retrieve data. Differently, the translation is between English and nRQL, while in our case between Romanian and SPARQL, where both nRQL and SPARQL are recognized by RacerPro.

Another notable effort in the context of Semantic Web is the combination between between ACE and GF from [15]. Approaches for verbalization based on ontology is introduced in [16] for English and Greek languages. A controlled natural language for editing ontology is presented in [17] based on Attempto Controlled English (ACE) language.

Linking tourism data. Regarding the link data component, a similar approach is the tourism linked data set in [18], based on the European statistics data from 1985 about 150 cities in Europe. The Lela system complements linked open data with information extracted from blogs to offer both subjective impressions about places and objective data. Besides DBpedia, YAGO2 [19] focuses on automatically extracting and publishing structured knowledge from Wikipedia. While the DBpedia taxonomy is manually developed and maintained, YAGO integrates the WordNet taxonomy, which leads to a higher number of classes in YAGO. Expanding our system to manage this richer taxonomy is one of the directions we intend to pursue as future work.

Romanian language processing. For the Romanian language several large annotated corpora do exist (George Orwell's novel 1984, Plato's Republic, ROCO), lexicons (WEB-DEX, CONCEDE, EUROVOC) [20] with the corresponding tools for exploiting these dictionaries (http://dexonline.ro/unelte) None of these resources deal with translation between a natural language and a formal language. We argue that such a translator can trigger various practical development at the application level. A Romanian grammar was developed by [21] that includes 866 grammatical rules and 320 affixes, which have been used for the development of a morphological vocabulary of cca. 30,000 words. For the natural language part of our work we based on the resource library for Romanian developed in [22]. Our morphological vocabulary was generated only for the tourism domain, with the goal to translate natural language queries into SPARQL. Our system for translating Romanian language queries into SPARQL syntax fills, in our view, an important gap among the existing linguistic resources for Romanian language [20].

VI. Conclusions

This paper introduced the Lela commercial product, which intends to be a semantic-based info-point for touristic information in Romania, offering both objective information and subjective impressions about places of interest. It provides data for the 4 main axes: accommodations, eating and drinkings, destinations and activities, with a special focus on the latter one. In order to provide these data, the system integrates Open Linked Data with subjective opinions expressed in articles to generate added value. The system also offers semantic search functionality through the Romanian natural language query interface, which translates the Romanian questions into SPARQL based on a controlled vocabulary derived from the developed LELA touristic ontology.

We are currently applying the natural language processing module to the task of populating the touristic objectives in Lela ontology with specific features identified in Romanian blogs.

The system is intended to be available for public use on http://www.lela.ro by the end of 2014.

Acknowledgements

Part of this work was supported by the PN-II Innovation Checks of the Romanian Ministry of Research for supporting research in small and medium enterprises, through the project "LELA - Collaborative Recommendation System in the Tourism Domain Using Semantic Web Technologies and Text Analysis in Romanian Language".

References

[1] Sutton, Charles and McCallum, Andrew, "An introduction to conditional random fields," vol., no., p., 267373, 2012. doi: 10.1561/2200000013. Available: http://dx.doi.org/10.1561/2200000013

[2] Bobicev, V. and Maxim, V and Prodan, T and Burciu, N. and Anghelus, V., "Emotions in words: developing a multilingual wordnet-affect," in *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics, Iasi, Romania*, 2010. doi: 10.1007/978-3-642-12116-6_31. Available: {http://dx.doi.org/10.1007/978-3-642-12116-6_31}

[3] Noy, Natalya F and McGuinness, Deborah L and others, "Ontology development 101: A guide to creating your first ontology," 2001.

[4] Pollock, Jeffrey T and Hodgson, Ralph, "Ontology design patterns," doi: 10.1002/0471714216.ch7. Available: http://dx.doi.org/10.1002/0471714216.ch7

[5] Baader, Franz, *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003. Available: http://dx.doi.org/10.2277/0521781760

[6] Haarslev, Volker and Hidde, Kay and Möller, Ralf and Wessel, Michael, "The racerpro knowledge representation and reasoning system," vol., no., 2012. doi: 10.3233/SW-2011-0032. Available: http://dx.doi.org/10.3233/SW-2011-0032

[7] Presutti, Valentina and Gangemi, Aldo, "Content ontology design patterns as practical building blocks for web ontologies," in *Conceptual Modeling-ER 2008*. Available: http://dx.doi.org/10.1007/978-3-540-87877-3_11

[8] Jens Lehmann and Robert Isele and Max Jakob and Anja Jentzsch and Dimitris Kontokostas and Pablo N. Mendes and Sebastian Hellmann and Mohamed Morsey and Patrick van Kleef and Sören Auer and Christian Bizer, "DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web Journal*, 2014.

[9] Watson, Mark, *Practical Semantic Web and Linked Data Applications - Common Lisp Edition*, 2010.

[10] Aarne Ranta, *Grammatical Framework: Programming with Multilingual Grammars*. Stanford: CSLI Publications, 2011, iSBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).

[11] Aarne Ranta, "Gf: A multilingual grammar formalism," vol., no., 2009. doi: 10.1111/j.1749-818X.2009.00155.x. Available: http://dx.doi.org/10.1111/j.1749-818X.2009.00155.x

[12] Lopez, Vanessa and Uren, Victoria and Sabou, Marta and Motta, Enrico, "Is question answering fit for the semantic web?: a survey," vol., no., 2011. doi: 10.3233/SW-2011-0041. Available: http://dx.doi.org/10.3233/SW-2011-0041

[13] Tablan, Valentin and Damljanovic, Danica and Bontcheva, Kalina, "A natural language query interface to structured information," in *The Semantic Web: Research and Applications*. Available: http://dx.doi.org/10.1007/978-3-540-68234-9_28

[14] Mithun, Shamima and Kosseim, Leila and Haarslev, Volker, "Resolving quantifier and number restriction to question owl ontologies," in *Semantics, Knowledge and Grid, Third International Conference on*. IEEE, 2007. Available: http://dx.doi.org/10.1109/SKG.2007.255

[15] Kaarel Kaljurand and Tobias Kuhn, "A multilingual semantic wiki based on attempto controlled english and grammatical framework," vol., abs/1303.4293, 2013. doi: 10.1007/978-3-642-38288-8_29. Available: http://dx.doi.org/10.1007/978-3-642-38288-8_29

[16] Androutsopoulos, Ion and Lampouras, Gerasimos and Galanis, Dimitrios, "Generating natural language descriptions from owl ontologies: the naturalowl system," vol., 2013. doi: 10.1613/jair.4017

[17] Kaarel Kaljurand, "ACE View — an ontology and rule editor based on Attempto Controlled English," in *5th OWL Experiences and Directions Workshop (OWLED 2008)*, Karlsruhe, Germany, 26–27 October 2008. doi: 10.5167/uzh-8822 12 pages. Available: http://dx.doi.org/10.5167/uzh-8822

[18] Sabou, Marta and Arsal, Irem and Braşoveanu, Adrian MP, "Tourmislod: A tourism linked data set," vol., no., 2013. doi: 10.3233/SW-2012-0087. Available: http://dx.doi.org/10.3233/SW-2012-0087

[19] Hoffart, Johannes and Suchanek, Fabian M. and Berberich, Klaus and Weikum, Gerhard, "YAGO2: A spatially and temporally enhanced knowledge base from wikipedia," vol., 2013. doi: 10.1016/j.artint.2012.06.001. Available: http://dx.doi.org/10.1016/j.artint.2012.06.001

[20] Cristea, Dan and Forăscu, Corina, "Linguistic resources and technologies for romanian language," vol., no., p., 40, 2006. doi: 10.1.1.414.9781

[21] Boian, E and Ciubotaru, C and Cojocaru, S and Colesnicov, A and Demidova, V and Malahova, L, "Lexical resources for romanian-a project overview," in *Symposium on Intelligent Systems and Applications, September*, 2003. Available: http://dx.doi.org/10.2218/jls.v1i1.824

[22] Enache, Ramona and Ranta, Aarne and Angelov, Krasimir, "An open-source computational grammar for romanian," in *Computational Linguistics and Intelligent Text Processing*, vol., ISBN 978-3-642-12115-9. Available: {http://dx.doi.org/10.1007/978-3-642-12116-6_14}